SatGen-CV: Semantic Image Synthesis for Satellite Imagery from Road and Building Segmentation Masks

Maxime Langelier
maxime.langelier@edu.devinci.fr

Corentin Juste
corentin.juste@edu.devinci.fr

Noé Le Yhuélic
noe.le_yhuelic@edu.devinci.fr

Vianney Le Bourhis
vianney.le_bourhis@edu.devinci.fr

Abstract

We present SatGen-CV, a Pix2Pix-based system for generating realistic satellite imagery from semantic segmentation masks that contain roads and buildings. Our approach addresses the practical challenge of generating synthetic large-scale aerial imagery by combining a U-Net generator with PatchGAN discriminator, multi-layer perceptual loss, and a seamless generation pipeline using cosine image blending. We train on the Massachusetts Roads Dataset and Inria Aerial Image Labeling Dataset, augmented with SAM-generated masks, achieving visually coherent synthesis. Our seamless generation method produces geographically consistent imagery without visible patch boundaries, enabling practical applications in urban planning simulation and synthetic data generation. We provide comprehensive implementation details, evaluate both quantitative metrics and qualitative results, and discuss limitations and future directions for this task.

1 Introduction

Generating photorealistic images from segmentation maps represents a fundamental challenge in computer vision with direct applications in urban planning and geospatial analysis. While substantial progress has been achieved in natural image domains through architectures like SPADE [2] and Pix2Pix [1], synthesizing satellite imagery from sparse semantic layouts presents unique challenges including maintaining geometric consistency across large areas and capturing domain-specific textures.

Generating realistic satellite imagery from road network and building footprint data addresses several critical applications. Urban planners can quickly visualize proposed developments before construction begins, enabling better decision-making and a faster process for new neighborhoods. The model could also be fine-tuned for some more specified satellite generation-related domains. Additionally, developers of autonomous systems (such as video games) can create simulation environments with realistic aerial imagery synthesized from map data.

Our work makes the following contributions:

1. A complete Pix2Pix-based pipeline specialized for satellite image synthesis from two-channel segmentation masks (roads and buildings).

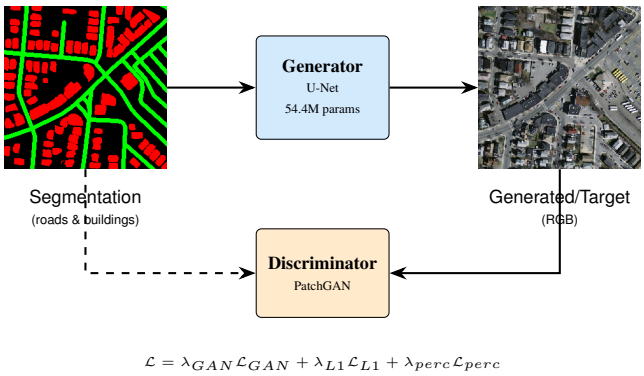

Figure 1: Overview of the SatGen-CV pipeline.

2. A seamless generation method using cosine-fade blending that produces arbitrarily large images without visible patch boundaries.

3. Integration with SAM (Segment Anything Model) [6] for automated building or road mask generation, allowing us to work with partially labeled datasets.

2 Related Work

2.1 Conditional Image Generation

Conditional GANs (cGANs) [3] extend the GAN framework by conditioning both generator and discriminator on auxiliary information (in our case, the original segmentation mask).

Pix2Pix [1] formalized paired image-to-image translation, combining a U-Net generator with skip connections and a PatchGAN discriminator. The architecture demonstrated compelling results on sattelite-to-map translation at 256×256 resolution, which is what motivated this project.

Pix2pixHD [4] extended to 2048×1024 through coarse-to-fine generation and multi-scale discrimination. However, computational costs increase substantially, motivating our focus on efficient patch-based processing.

2.2 Spatially-Adaptive Normalization

SPADE [2] addressed semantic information loss in normalization layers by modulating parameters spatially according

to the input layout:

$$\text{SPADE}(h, m) = \gamma(m)\frac{h - \mu}{\sigma} + \beta(m) \tag{1}$$

where learned functions γ and β produce spatially-varying parameters from the semantic map m. While SPADE achieves superior semantic correspondence, its increased complexity motivated our choice of the simpler Pix2Pix baseline for this initial implementation.

2.3 Perceptual Loss Functions

Beyond pixel-wise reconstruction, perceptual losses [10] compare features from pretrained networks. Multi-layer feature matching at different VGG depths captures both low-level textures and high-level semantics, improving visual quality over L1 loss alone. We chose to experiment with perceptual loss in this project to see if we can achieve decent performance with it, although it was not present in the base Pix2Pix implementation.

3 Methodology

3.1 Problem Formulation

Given a two-channel semantic segmentation mask S of shape $H \times W \times 2$ containing binary building footprints and road networks, we aim to generate a photorealistic RGB satellite image I consistent with the input layout. We formulate this as a conditional image generation problem, learning a mapping $G : S \to I$ that minimizes a combination of adversarial, reconstruction, and perceptual losses.

3.2 Network Architecture

3.2.1 Generator

We employ a U-Net architecture with skip connections [11], modified for our input-output specification. Similar to the base implementation of Pix2Pix, the encoder consists of 8 downsampling blocks, each containing:

- Convolution (kernel=4, stride=2, padding=1)
- Instance Normalization [12]
- LeakyReLU activation ($\alpha = 0.2$)
- Dropout (rate=0.5) in early layers

The decoder mirrors this structure with transposed convolutions, ReLU activations, and skip connections from corresponding encoder layers. Feature dimensions progress as: $64 \to 128 \to 256 \to 512 \to 512 \to 512 \to 512 \to 512$ in the encoder, with symmetric expansion in the decoder. The final layer applies Tanh activation, producing outputs in $[-1, 1]$. We then scale these back to $[0, 256]$ to get the output image.

The generator contains approximately 54.4 million parameters. We use Instance Normalization rather than Batch Normalization, as it provides better results for image generation tasks where batch statistics are less meaningful.

3.2.2 Discriminator

We use a PatchGAN discriminator [1] that classifies whether overlapping image patches are real or fake, rather than producing a single scalar output. The idea behind this is to encourage high-frequency detail and to reduce parameter count compared to full-image discrimination.

The discriminator receives concatenated input $[S, I]$, which results in shape $H \times W \times 5$. This is a key difference with the base Pix2Pix implementation which is of shape $H \times (2 \times W) \times 3$, something we couldn't do since we had only two input channels for three output channels.

3.3 Loss Functions

Our total generator loss combines three components:

$$\mathcal{L}_G = \lambda_{GAN}\mathcal{L}_{GAN} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{perc}\mathcal{L}_{perc} \tag{2}$$

3.3.1 Adversarial Loss

We use binary cross-entropy for the adversarial objective:

$$\mathcal{L}_{GAN} = \mathbb{E}[\log(1 - D(S, G(S)))] \tag{3}$$

For stable training, we apply label smoothing by setting real targets to 0.9 rather than 1.0.

3.3.2 L1 Reconstruction Loss

Pixel-wise L1 loss encourages structural consistency:

$$\mathcal{L}_{L1} = \mathbb{E}[\|I_{target} - G(S)\|_1] \tag{4}$$

3.3.3 Multi-Layer Perceptual Loss

We extract features from five layers of a pretrained VGG-19 network [14]:

$$\mathcal{L}_{perc} = \sum_l w_l \|\phi_l(I_{target}) - \phi_l(G(S))\|_1 \tag{5}$$

where ϕ_l denotes features at layer l (relu1_2, relu2_2, relu3_4, relu4_4, relu5_4) and weights $w_l = \{1/32, 1/16, 1/8, 1/4, 1\}$ emphasize deeper semantic features.

3.3.4 Discriminator Loss

$$\mathcal{L}_D = \frac{1}{2}\mathbb{E}[\log D(S, I_{real})] + \frac{1}{2}\mathbb{E}[\log(1 - D(S, G(S)))] \tag{6}$$

3.4 Seamless Large-Scale Generation

An important feature to this project is our seamless generation pipeline for producing arbitrarily large satellite images without visible patch boundaries.

3.4.1 Patch Grid Computation

Given image dimensions (H, W), patch size p, and overlap o, we compute patch positions ensuring complete coverage. In our implementation, we set $p = 256$ and $o = 128$

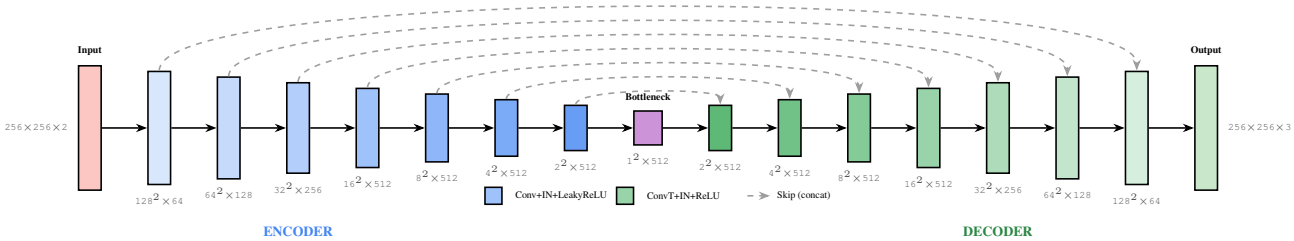

Figure 2: Generator architecture. A U-Net encoder-decoder with skip connections processes the 2-channel segmentation input to produce 3-channel RGB output. The encoder progressively downsamples while increasing feature depth (blue), and the decoder upsamples with skip connections from corresponding encoder layers (green). Dropout is applied in the first three decoder blocks.

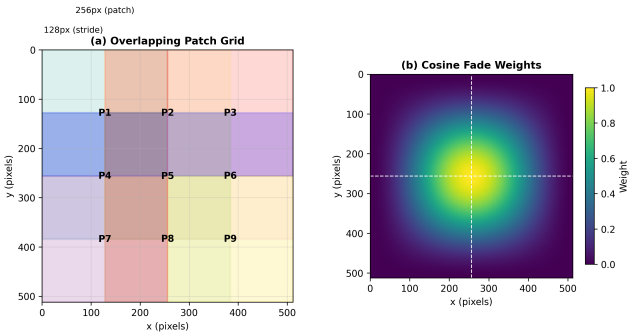

Figure 3: Seamless generation via cosine-fade blending. (a) Overlapping patch grid covering the input ($o = 128$). (b) Blend weights from cosine functions.

3.4.2 Cosine-Fade Blending

We blend overlapping patches using smooth cosine transitions. For overlap region width o, we define two fade functions:

$$f_{in}(t) = \frac{1}{2}(1 - \cos(\pi t)), \quad t \in [0, 1] \tag{7}$$

$$f_{out}(t) = \frac{1}{2}(1 + \cos(\pi t)) \tag{8}$$

The 2D blend weights are computed by the product of their cosine profiles, creating smooth transitions in both dimensions. Final pixel values are then weighted averages:

$$I_{final}(x, y) = \frac{\sum_k w_k(x, y) \cdot P_k(x, y)}{\sum_k w_k(x, y)} \tag{9}$$

where P_k are overlapping patches and w_k their blend weights.

We also experimented with linear and gaussian blending techniques, and retained the cosine one as it gave the best results.

3.4.3 Noise Injection for Empty Regions

Empty segmentation patches will produce identical outputs, creating visible repetition artifacts for images with large blank regions. We inject Gaussian noise on empty patches before generation, both during training and during inference. This introduces stochasticity to help produce varied background textures on blank patches.

3.4.4 Inference-Time Dropout

Following the implementation on the original Pix2Pix paper, we keep dropout enabled during inference. This adds controlled randomness, preventing the generator from producing overly deterministic outputs for similar inputs.

4 Experiments

4.1 Dataset

4.1.1 Massachusetts Roads Dataset

We train on the Massachusetts Roads Dataset [7], containing 1,171 aerial images at 1500×1500 pixels. Each image includes binary road segmentation masks derived from OpenStreetMap. The dataset presents a landscape that is highly repetitive, with recurring patterns of residential areas, similar road network configurations, and consistent architectural styles across all the dataset.

4.1.2 Inria Aerial Image Labeling Dataset

To help mitigate the relative monotony of the Massachussets Roads Dataset, we additionally train on the Inria Aerial Image Labeling Dataset [8], which provides high-resolution aerial imagery covering diverse urban landscapes across five cities: Austin, Chicago, Kitsap County, Western Tyrol, and Vienna. Each image is 5000×5000 pixels with corresponding binary building footprint annotations, but we rescale it to 1500×1500 for scale consistance. This dataset complements Massachusetts Roads by providing building-focused labels and geographic diversity, exposing the model to much more varied architectural styles and urban densities.

4.1.3 Building Mask Generation with SAM3

Road masks alone provide few supervision to the model. Besides, Both our datasets have masks representing different features (roads and buildings respectively). To fix this, we augment our Massachussets Roads Dataset with building footprints, and our Inria Aerial Image Labeling Dataset with road masks by using segmentation masks generated using the Segment Anything Model (SAM) [6]. SAM's zero-shot segmentation capability enables automated road and building detection without task-specific training, creating complementary two-channel inputs that fit our use case.

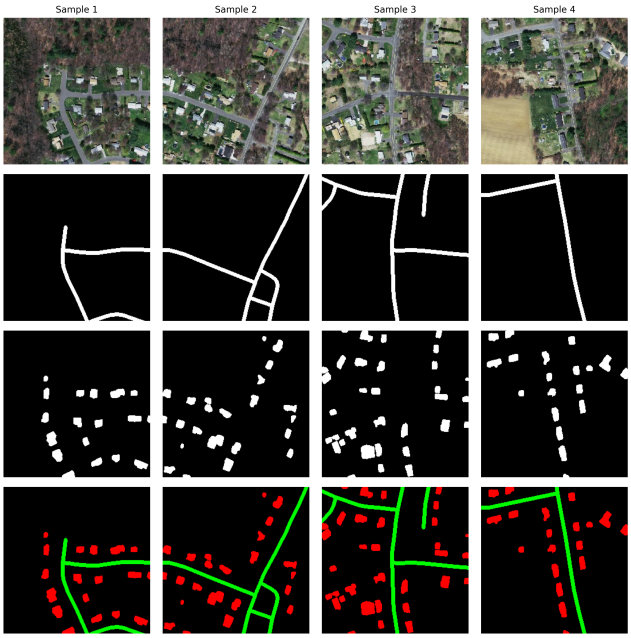

Figure 4: Dataset preparation. Original satellite images (top) with corresponding road masks from OpenStreetMap and SAM-generated building footprints, combined into two-channel segmentation inputs.

4.1.4 Data Preprocessing

We apply the following preprocessing pipeline:

1. **Patch extraction**: 256×256 patches with configurable stride (256 for non-overlapping, 128 for 50% overlap).
2. **Quality filtering**: Remove patches where satellite images contain >5% white pixels (blank areas) and segmentation masks contain <2% non-zero pixels (empty labels).
3. **Normalization**: Scale pixel values to be in the range $[-1, 1]$.

After filtering, approximately 70-80% of patches are retained, ensuring the model trains on informative samples only. This means we are left with a training set of about 80.000 training patches.

4.1.5 Data Augmentation

During training, we apply the following augmentation techniques:

- Random horizontal flip
- Random vertical flip
- Random 90°/180°/270° rotation
- Slight color shift on target images (brightness, contrast, saturation ±10%)

Color shift is applied only to target images, not on the input segmentation masks.

Table 1: Training Hyperparameters

Parameter	Value
Batch size	1–8
Epochs	10
Optimizer	Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$)
Learning rate (G)	0.0002–0.0003
Learning rate (D)	0.0002–0.0003
LR scheduler	CosineAnnealing ($\eta_{min} = 10^{-6}$)
λ_{GAN}	1.0
λ_{L1}	10
λ_{perc}	50
Patch size	256×256

4.2 Training Configuration

Table 1 summarizes our training hyperparameters.

We train using Adam optimizer with $\beta_1 = 0.5$ for both generator and discriminator. The learning rate follows cosine annealing.

4.3 Training Procedure

Each iteration proceeds as:

1. **Discriminator update**: Generate fake images, compute discriminator loss on real (target=1) and fake (target=0) pairs, backpropagate.
2. **Generator update**: Generate new fake images, compute combined loss $\mathcal{L}_G$, backpropagate.

4.4 Evaluation Metrics

We evaluate using:

- **L1 Loss**: Pixel-wise reconstruction error on held-out test set.
- **Perceptual Loss**: VGG feature distance indicating semantic similarity.
- **Visual Inspection**: Qualitative assessment of visual consistency, texture realism, and seamless generation quality.

4.5 Implementation Details

We implement our model in PyTorch with the following specifications:

- Hardware: Training Apple Silicon MPS GPU (30+ hours)
- VGG-19 pretrained on ImageNet for perceptual loss
- TensorBoard logging at every batch
- Checkpoint saving every epoch with best model tracking
- Validation images logged for qualitative monitoring

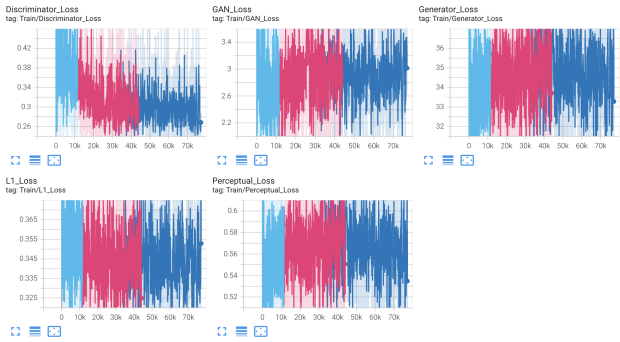

Figure 5: Training curves. Discriminator loss decrease steadily. to 0.25. Generator loss stagnates at arount 35, indicating it is the bottleneck here.

5 Results

5.1 Quantitative Results

We can see the validation metrics of our GAN in the table below:

Metric	Value
Generator Loss	34.1628
Discriminator Loss	0.1881
L1 Loss	0.3388
Perceptual Loss	0.5550

Table 2: Validation metrics. Discriminator loss on the validation set appears even lower than on the training set (0.18)

5.2 Training Dynamics

Figure 5 shows training progression. Discriminator loss decreases consistently to about 0.25 while generator loss stays relatively stable arount 35, indicating our discriminator learns much faster than our generator. We have tried several changes to this, such as manually decrasing the learning rate during training, but our generator's simple architecture remains the main bottleneck here.

5.3 Qualitative Results

Figure 6 presents qualitative results on test samples. Key observations:

- **Road continuity**: Generated roads maintain geometric consistency and appropriate width.
- **Building textures**: Building regions exhibit varied rooftop appearances reflecting the diverse training data from Massachusetts and European cities.
- **Background synthesis**: Vegetation and terrain textures appear naturally in some places despite the lack of dedicated segmentation masks.

5.4 Seamless Generation Results

Figure 7 demonstrates seamless generation at 1408×1408 resolution. The cosine-fade blending successfully eliminates patch boundaries while maintaining consistent appearance. Roads crossing patch boundaries remain continuous, and vegetation textures blend smoothly.

5.5 Blending Methods

Figure 8 compares blending strategies, demonstrating the effectiveness of cosine-fade blending for seamless large-scale generation.

6 Discussion

6.1 Strengths

Our approach has several strengths:

- **Ease of use and deployment**: The combination of efficient Pix2Pix architecture with seamless generation enables practical applications without the need for high-end hardware.
- **Data flexibility**: SAM integration for building mask generation removes the need for fully pre-annotated datasets, enabling deployment across diverse regions.
- **Scalability**: Patch-based processing with seamless blending allows for the generartion of arbitrarily large images, addressing a key limitation of fixed-size models.

6.2 Limitations

Despite the strengths of this apporach, several limitations exist:

- **Input sparsity sensitivity**: Very sparse inputs (isolated road segments) can produce inconsistent surrounding context, as the model lacks global priors about expected terrain. Indeed some repeated artifacts remain and would need to be addressed.
- **Limited control**: The model learns an averaged style from training data, limiting control over specific architectural or vegetation characteristics.
- **Resolution constraints**: 256×256 patch size limits fine detail; higher resolutions would require architectural modifications and a higher resolution dataset.
- **Domain gap**: While training on both Massachusetts and Inria datasets improves generalization, the model may still struggle with geographic regions significantly different from the training distribution.
- **Data quality**: As can be seen in Figure 6, SAM can sometimes fail to produce coherent segmentation masks for roads and buildings. This is particularly noticeable for the Inria Aerial Image Dataset, where road masks quality is highly inconsistent. Our training data suffers from this. Better data annotation could lead to improved performance.

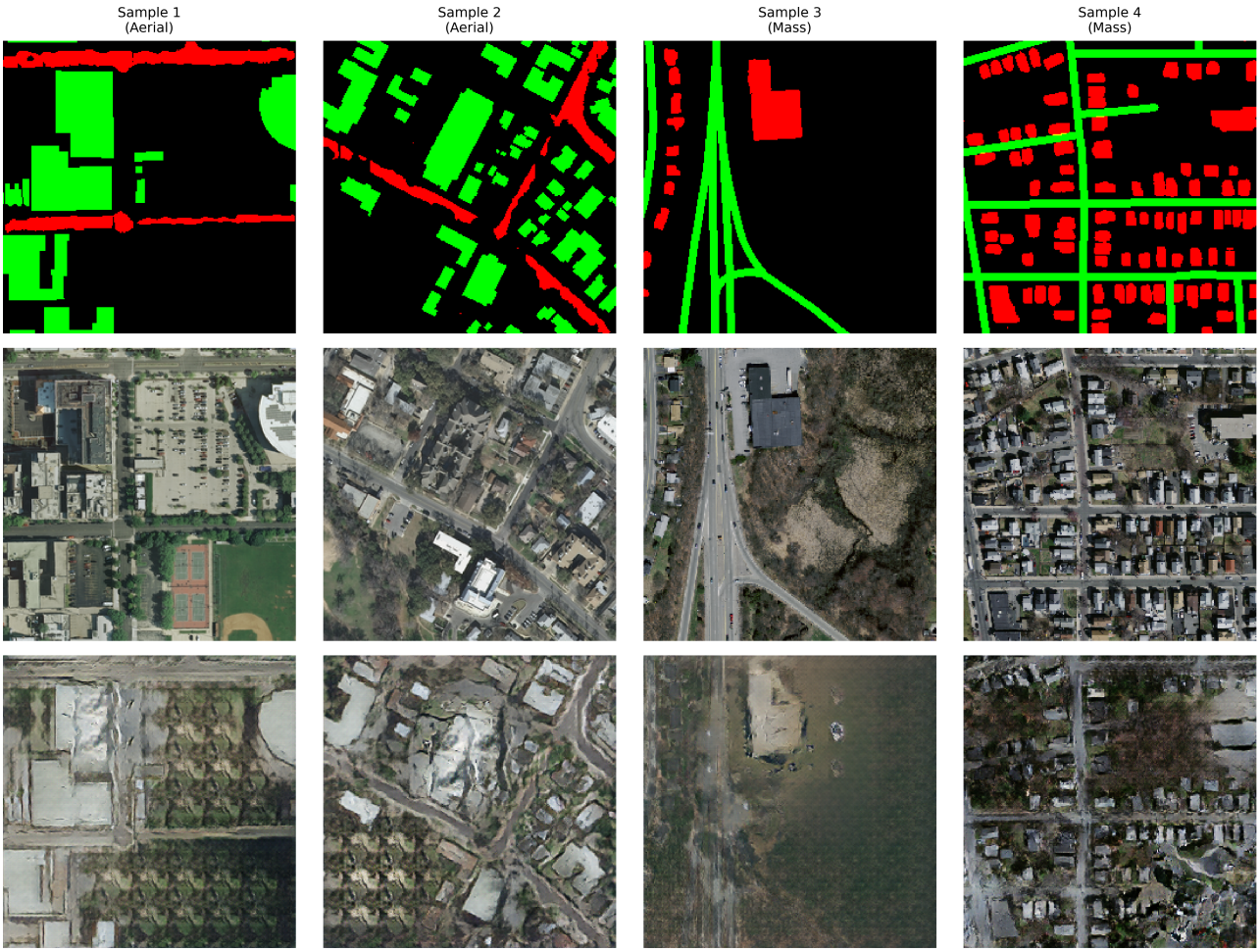

Figure 6: Qualitative results on test samples. Each column shows a different test sample from the Inria Aerial Image Labeling or Massachusetts Roads datasets. From top to bottom: input segmentation mask (roads in red, buildings in green), ground truth satellite image, and generated output.

Figure 7: Seamless generation of a 1408×1408 image from 256×256 patches with 64-pixel overlap. Cosine-fade blending eliminates visible boundaries while maintaining consistent lighting and texture across the image.

Figure 8: Comparison of blending strategies for seamless generation. Direct stitching creates obvious seams at patch boundaries. Linear blending reduces artifacts but leaves subtle discontinuities. Cosine-fade blending produces the smoothest results by matching derivatives at overlap boundaries.

6.3 Lessons Learned

Key insights from development:

1. High perceptual loss weight ($\lambda_{perc} = 50$) proved essential for texture quality, and was a great improvement over the standard Pix2Pix implementation.

2. Maintaining dropout at inference time, counterintuitively, improves output diversity and reduces artifacts on sparse inputs.

3. 64-pixel overlap (25% of patch size) provides the minimum necessary for seamless blending; smaller overlaps produce visible artifacts.

6.4 Future Directions

Several extensions could improve our system:

1. **Better data preprocessing**: Besides basic quality filtering, relatively few data preprocessing was done on the dataset to ensure diversity and quality of the inputs provided.

2. **Attention mechanisms**: Cross-patch attention could maintain feature consistency without explicit blending.

3. **Diffusion models**: Recent diffusion-based approaches may offer improved quality and diversity.

4. **Better dataset**: Extending our dataset by using sattelite imagery from third party providers could help us gather more diverse training data, yielding better results.

7 Conclusion

This paper introduced SatGen-CV, a practical system for generating satellite imagery from road and building segmentation masks.

Key contributions include the seamless generation pipeline eliminating patch boundary artifacts, noise injection strategies for consistent background synthesis, and comprehensive documentation enabling reproducibility and future contributions. Quantitative and qualitative evaluations demonstrate effective synthesis on the Massachusetts Roads and Inria datasets.

While limitations exist in input sparsity handling and style control, our work provides a solid foundation for infrastructure-focused satellite image synthesis. The practical importance of synthetic satellite imagery for urban planning, simulation work, and data augmentation ensures real world applications subsist.

Code and pretrained models are available at: `[https://github.com/Guereak/SatGen-CV]`, `[https://huggingface.co/Guereak/SatGen-CV]`.

References

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE CVPR*, 2017.

[2] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE CVPR*, 2019.

[3] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784*, 2014.

[4] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE CVPR*, 2018.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE ICCV*, 2017.

[6] A. Kirillov *et al.*, "Segment anything," in *Proc. IEEE ICCV*, 2023.

[7] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, Univ. Toronto, Toronto, ON, Canada, 2013.

[8] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The Inria aerial image labeling benchmark," in *Proc. IEEE IGARSS*, 2017.

[9] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE CVPR*, 2018.

[10] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*, 2016.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015.

[12] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv:1607.08022*, 2016.

[13] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. ICLR*, 2018.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.

[15] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, 2003.

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NeurIPS*, 2020.